

**REMARKS**

Claims 1-24 are pending in this application. Claim 17 has been amended by this Amendment.

The Office Action dated August 27, 2002 rejected claims 1-2, 9-10 and 17-18 under 35 U.S.C. §102(b) as being anticipated by prior art and rejected claims 3-8, 11-16 and 19-24 under 35 U.S.C. §103 as being obvious in view of prior art.

**Claim Rejections - 35 U.S.C. §102(b)**

The grounds for the rejections of claims 1-2, 9-10, and 17-18 under 35 U.S.C. §102(b) as being anticipated by prior art is set forth on pages 2-3 of the Office Action. Specifically, the rejections rely upon the multiprocessor system described in U.S. Patent No. 4,543,627 issued to Schwab (this multiprocessor system hereinafter being referred to simply as “Schwab”). Applicant respectfully traverses the rejections and submits that the rejections fail to establish a prima facie case that Schwab contains each and every one of the combination of features recited in the rejected claims.

For example, independent claims 1, 9 and 17 recite that a message sent to a remote device is processed “...in said remote device, to determine whether or not the transport header of said message identifies the message as a type of remote Direct Memory Access (rDMA) read operation” and that “if the remote device determines from the transport header of said message that the message is said type of remote Direct Memory Access (rDMA) read operation, then performing a remote Direct Memory Access (rDMA) write operation to said local device in accordance with data elements included in said message.” The rejections

Application No. 09/397,850

identify various passages in the '627 patent in conjunction with these features.

The rejection identifies the following passages in conjunction with the feature that a message sent to a remote device is processed "...in said remote device, to determine whether or not the transport header of said message identifies the message as a type of remote Direct Memory Access (rDMA) read operation":

"...Each of the processor interface units 100-107, uses one of the direct memory access units 200-207 to communicate with processor 300, and uses one of the direct memory access units 400-407 to communicate with the associated auxiliary processor. For the sake of convenience, in this description a phrase such as: "peripheral interface unit 100 reads from, writes into, changes, or updates processor 300 or 500" should be interpreted as a brief way of saying that peripheral interface unit reads from, writes into, changes, or updates processor 300 or 500 using direct memory access 200 or 400, respectively." (col. 3, lines 25-36)(underlining added)

"FIG. 3 shows the format of the message transmitted between processor 300 and processor 500. Every message has a standard format initial header 351, and an arbitrary length body 360. The header has a number of fields which are fixed in location in every message, and are interpreted by the receiving PIUMH and the processor interface unit. The fields are used to accomplish a number of purposes. The "from" field 352 is the identification of the sending process number. The "to" field 353 is the identification of the receiving process number. The "message size" field 354 is an indication of the total length of the message including both header and body. The "type of message" field 355 is an indication of whether this is, for example, a simple message (MSG), or one of the messages which are used to prepare for the transmission of a block of data. The "status" field 356 is used to indicate the nature of a failure. The "identity" field 359 identifies one of a series of messages from different sources, in order to allow any responses to be associated with the proper source. The "block size" field 362 indicates the length of a block to be transferred when a block transfer is requested." (col. 6, lines 47-68)

"The processor interface unit plays a much broader role in the case of block transfer than it does for the transfer of simple messages. For the block transfer, also illustrated in FIG. 2, the processor interface unit must recognize that it is to read information, not only from the send buffers of the sending processor, but from an address indirectly specified by one of the messages. The processor interface unit examines the message type field 355 of the header 351 of every message to see if the message involves the preparation for a block transfer. When this type of message is recognized, the processor interface unit must find the address of the sending and receiving processor message blocks in order to carry out the block transfer." (col. 7, lines 37-50)(underlining added)

"The message contains the sending job buffer number 630 in the send job buffer field 357, and indicates the nature of the block transfer request in the type field 355. In this case, a "write" (WT) message, i.e., a message to initiate a write block transfer, is sent in the type field 355." (col. 8, lines 1-5)

The cited passage at column 3 of the patent indicates that there is a central processor 300 and a plurality of processors 500-507, and that a peripheral interface unit 100 reads from or writes into the processors using direct memory access 200 or 400 shown in Fig. 1 of the patent. As indicated here and elsewhere in the patent, such as in the cited passage in column

Application No. 09/397,850

7, it is peripheral interface unit 100 which processes the messages sent between the processors to conduct data transfers. Thus, independent claims 1, 9 and 17 are not anticipated by Schwab because, in Schwab, it is not the remote device (processor) which processes the messages as recited in this feature of independent claims 1, 9 and 17.

The passage at column 6 of the patent indicates that the header of messages passed between the processors includes a “type of message” field 355. Field 355 indicates, for example, whether the message is one of two categories: 1) a simple message (MSG); or 2) one of the messages which are used to prepare for the transmission of block of data. The block data transfer is addressed by the cited passage at column 7 of the patent. The cited passage at column 8 of the patent states that field 355 in the header can be processed to determine whether a block data transfer is a read operation or a write operation. While field 355 in the header may be used to identify a message as a read operation or a write operation, applicant can find nothing which suggests that field 355 can be used to identify the “type of” remote Direct Memory Access read operation as recited in the claims, in particular the type of operation that is recited in the second above-quoted feature of the claims.

The second above-quoted feature of the claims is that “if the remote device determines from the transport header of said message that the message is said type of remote Direct Memory Access (rDMA) read operation, then performing a remote Direct Memory Access (rDMA) write operation to said local device in accordance with data elements included in said message.” Thus, the “type of” remote Direct Memory Access read operation is one which causes the remote (receiving) device to perform “a remote Direct Memory Access (rDMA) write operation to said local device in accordance with data elements included in said message”. In other words, the recited type of read operation is not a conventional read

Application No. 09/397,850

operation - it is a read operation in which the remote (receiving) device performs a write operation to the local (sending) device. The rejections paraphrase this feature and do not explicitly acknowledge that the type of read operation causes the remote device to perform a write operation. In addition to some of those previously cited, the rejections rely on the following passages of the '627 patent:

The central processing unit 112 in processor interface unit 100 controls all of the transmission of messages from processor 300 to processor 500 and vice versa. Processor interface unit 100 has access to the memory of processor 300 via direct memory access 200 and uses this access to read messages from and to write messages into processor 300. The messages from processor 300 memory 310 are read into processor interface unit 100 memory 111, and the messages destined for processor 300 are written from processor interface unit 100 memory 111 into processor 300 memory 310. Similarly, messages destined for processor 500 are written from the processor interface unit 100 memory 111 via the direct memory access 400 into processor 500 memory 510, and messages from processor 500 to processor 300 are read from processor 500 memory 510 by the direct memory access 400 into processor interface unit 100 memory 111. (col. 4, lines 46-63)

The processor interface unit can also be used to transfer a block of data from an arbitrary location in memory of one processor to an arbitrary location in memory of another processor. If the data is transferred to the processor making the request, this is called a read block transfer; if it is transferred from the processor making the request, this is called a write block transfer. Such a transfer of information can be processed more efficiently as a block transfer than as a series of data messages, each containing a portion of the block of data to be transferred. (col. 7, lines 15-25)

Each processor involved in the block transfer will assign a special job buffer to help control this transfer. Job buffer 630 in processor 300 and job buffer 730 in processor 500 contain the address entries 631 and 731, and size entries 632 and 732, of the transmitted or received block of information, the status entries 633 and 733 of this block transfer, and the number of the job buffer in the other processor 634 and 734. The processor interface unit 100 reads the sending (357) and receiving (358) job buffer number fields which define the addresses of these job buffers, and then reads the job buffers in order to find the initial address, the size of the block to be transmitted, and the destination address of the block. (col. 7, lines 51-64)

These cited passages indicate that processor interface unit 100 performs the direct memory access read and write operations. Apart from the simple fact that the processor interface unit 100 performs the operations, there is nothing in the cited passages which suggest the type of read operation recited in the claims in which the remote device performs a "a remote Direct Memory Access (rDMA) write operation to said local device in accordance with data elements included in said message". The read and write operations discussed in the cited passages appear to be conventional read and write operations.

Application No. 09/397,850

**Claims Rejections - 35 U.S.C. §103**

The grounds for the rejections of dependent claims 3-8, 11-16 and 19-24 under 35 U.S.C. §103 as being rendered obvious by prior art is set forth on pages 3-7 of the Office Action. Specifically, the rejections rely upon a number of patents in addition to U.S. Patent No. 4,543,627. Applicant respectfully traverses the rejections and submits that the rejections fail to establish a prima facie case that each and every one of the combination of features recited in the rejected claims is suggested by the cited patents.

As an example, claims 3, 11 and 19 recite a Virtual Interface network interface controller (each one of the remaining rejected claims is directly or indirectly dependent on one of claims 3, 11 and 19). The rejections rely upon “Osborne” for this feature<sup>1</sup>, noting that it uses a virtual addresses for data transfer. However, the rejection apparently does not appreciate the Virtual Interface (VI) involves more than the use of virtual addresses. Applicant points to the discussion of the VI architecture in the originally filed specification (see page 17, line 11, to page 18, line 16). Indeed, the patents relied upon in the obviousness rejection were filed before the VI architecture was published in 1997.

Applicant submit that none of claims 3-8, 11-16 and 19-24 are rendered obvious by the cited references at least for this reason and requests that the rejection be withdrawn.

**Claim Amendments**

Claim 17 has been amended by this Amendment merely to insert the word “to” in the preamble in order to correct a grammatical error discovered in the preamble. Applicant respectfully submits that the amendment does not change the substantive scope of claim 17.

Application No. 09/397,850

To the extent necessary, Applicants petition for an extension of time under 37 CFR §1.136. Please charge any shortage in the fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account No. 01-2135 (Case No. 219.37206X00) and please credit any excess fees to such deposit account.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

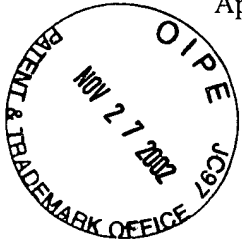
A handwritten signature in black ink, appearing to read "Robert M. Bauer". The signature is fluid and cursive, with the first name "Robert" and last name "Bauer" clearly distinguishable.

Robert M. Bauer, Registration No. 34,487

1300 North Seventeenth Street  
Suite 1800  
Arlington, VA 22209  
Tel: 703/312-6600  
Fax: 703/312-6666

---

1 The rejection is ambiguous because the Office Action cites two patents issued to Osborne and the rejection does not specify which one of the two patents is relied upon in the rejection.



**Version with Markings to Show Changes Made**

17. A tangible medium storing a plurality of program instructions, said program instructions when executed by a processor causing a network device to read data from a remote memory of a remote device directly to a local memory of a local device by:

- sending a message from said local device to said remote device, said message including a transport header indicating the message type of said message;
- processing said message, in said remote device, to determine whether or not the transport header of said message identifies the message as a type of remote Direct Memory Access (dram) read operation; and
- if the remote device determines from the transport header of said message that the message is said type of remote Direct Memory Access (dram) read operation, then performing a remote Direct Memory Access (dram) write operation to said local device in accordance with data elements included in said message;
- sending a message from said local device to said remote device, said message including a transport header indicating the message type of said message;
- processing said message, in said remote device, to determine whether or not the transport header of said message identifies the message as a type of remote Direct Memory Access (dram) read operation; and

Application No. 09/397,850

if the remote device determines from the transport header of said message that the message is said type of remote Direct Memory Access (dram) read operation, then performing a remote Direct Memory Access (dram) write operation to said local device in accordance with data elements included in said message.